In the Claims:

1. (previously presented) A method of implementing musical instrument digital interface (MIDI) synthesis comprising the steps of:

(a) providing a wireless handset having dual processor management control associated with a general purpose processor (GPP) and a digital signal processor (DSP), a flash memory having MIDI data stored therein, a digital-to-analog converter (DAC) and digital signal processor (DSP) peripherals to drive the DAC;

(b) interrogating the flash memory via the GPP to open a MIDI bit stream and determine sample sets to be loaded into a DSP memory associated with the DSP;

(c) loading and instantiating via the GPP, a DSP code associated with the sample sets into the DSP memory;

(d) initializing a sample set memory and signaling the DSP to start running a DSP synthesizer;

(e) parsing the MIDI bit stream into synthesis packets comprising MIDI commands via the GPP;

(f) transferring the synthesis packets to the DSP via the GPP; and

(g) time stamping and synthesizing the MIDI commands via the DSP to render audio signals to the DAC.

2. (previously presented) The method according to claim 1 further comprising the steps of:

(h) closing the MIDI bit stream when the MIDI bit stream has been exhausted;

(i) causing the DSP to stop synthesizing the MIDI commands; and

(j) de-allocating the sample set memory.

3. (previously presented) The method according to claim 1 further comprising the step of closing the MIDI bit stream in response to a user command.

4. (previously presented) A method of implementing musical instrument digital interface (MIDI) synthesis comprising the steps of:

(a) providing a wireless handset having dual processor management control associated with a first data processor and a second data processor, a flash memory having MIDI data stored therein, a digital-to-analog converter (DAC) and digital signal processor (DSP) peripherals to drive the DAC;

(b) interrogating the flash memory via the first data processor to open a MIDI bit stream and determine sample sets to be loaded into a shared memory;

(c) loading and instantiating via the first data processor, a second data processor code associated with the sample sets into the shared memory;

(d) initializing a sample set associated with the shared memory and signaling the second data processor to start running a MIDI synthesizer;

(e) parsing the MIDI bit stream into synthesis packets comprising MIDI commands via the first data processor;

(f) transferring the synthesis packets to the second data processor via the first data processor; and

(g) time stamping and synthesizing the MIDI commands via the second data processor to render audio signals to the DAC.

5. (previously presented) The method according to claim 4 further comprising the steps of:

(h) closing the MIDI bit stream when the MIDI bit stream has been completely read;

(i) causing the second data processor to stop synthesizing the MIDI commands; and

(j) de-allocating the sample set memory.

6-10 (canceled)